Ｉ／Ｏルーチン作成支援ツールについて

(株) システムエンジニアリング　○正会員　北島　彰夫
正会員　越後　　滋

１．まえがき

最近パソコンが急激に普及しているが、その１つの要因として、日本語処理，グラフィック機能が充実しており、これらを利用した対話型式的処理を比較的簡単にできるといったことが挙げられよう。しかし、この対話処理を取入れたアプリケーションソフトは利用者には大変便利であるものの、その開発者にとっては大変な労力をしいられるものである。すなわち、利用者側の種々のステージに応じた適切な画面表示を行ない、入力示方書とか取扱説明書とかがなくても、誤りなく入力、処理が行なえるものとしなければならない。そのため画面にコメント表示ばかりでなく、場合によってはグラフィック表示も必要となる。また入力の際には、誤操作、誤入力が必ずあることを想定して、これらに対処できるようにしておく必要もある。

筆者らは土木関係のアプリケーションソフトの開発を手がけてきたが、プログラムの６～７割が入出力部分といったものも多く、これら入出力部分の開発の手間は膨大なものとなる。しかも、こういった入出力部分はアプリケーションそのものの本質部分（たとえば構造解析、応力計算等）とは異なり、それ程高度な専門知識は必要としない。そこで、こういった対話型機能を有する入出力ルーチンをわざわざＢＡＳＩＣ文法にのっとってプログラマがコーディングしなくても、画面をみながら簡単なキー操作を行なうことで、自動的に生成する開発支援ツールＩＯＴＯＯＬ（ＩＮＰＵＴ　ＯＵＴＰＵＴ　ＴＯＯＬ）を開発したので、その概要について報告する。

２．基本仕様

本ツールの特記すべき基本示様は以下の通りである。

a.　本ツールの管理下で、対話処理により入出力ルーチンで実現させたい画面を、ＣＲＴ画面上に直接作る。作るものは文字（漢字を含む）、表，図形，入出力変数とその表示エリア（場所，桁数，小数点位置）等とする。

b.　これらの画面上に定義した文字，図形，変数等は自由に変更，修正，削除，行換え等の編集ができるものとする。

c.　画面作成が終了したら、その画面を実現するＢＡＳＩＣプログラムを自動生成する。作られたプログラムには、キー入力，入力値の画面表示，入力値の修正，データファイル（インデックスシーケンシャルファイル）への書込み，読出しの基本ルーチンがすべて組込まれるものとする。

d.　いったん作られたプログラムに対して、本ツールにより再編集が行なえるとともに、直接人手により手直しすることもできるものとする。

e.　２５行×８０カラムの画面（パソコンにより異なる場合もある）の２５行目をガイダンスエリアとして用いる。このエリアに表示される情報のみで、誰でもが簡単に操作できるものとする。

f.　操作性を増大させるためにキー操作はすべて、ワンタッチ方式とし、ＲＥＴＵＲＮキーを押す必要はない。また対話処理中、操作ミスをしたらＲＥＴＵＲＮキーを押し続ければ、元の状態に復帰できるものとする（ミスを犯したコマンドからのぬけ出し）。

３．システム構成

本ツールは図－１に示す７つのルーチン群によって構成されている。

(1)　基本入力ルーチンでは、作成するプログラムの収容ファイル名、プログラムの新規作成または修正の区

分、開始文番号とそのピッチ、プログラムタイトル、プログラムの命令形式（たとえばＰＲＩＮＴ文，ＰＲＩＮＴ　ＵＳＩＮＧ文のいずれかの選択）等の基本データの入力を行なう。

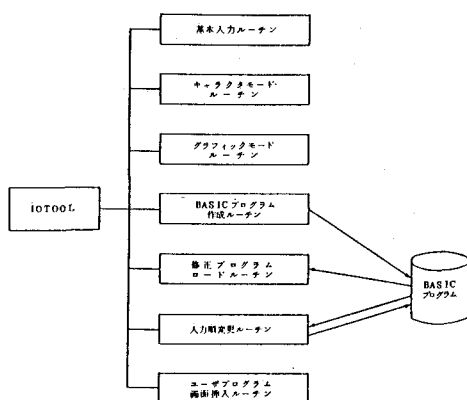
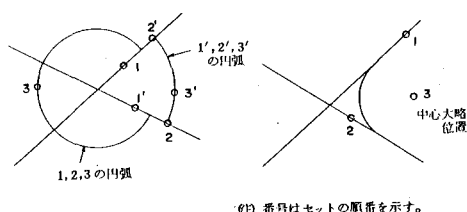
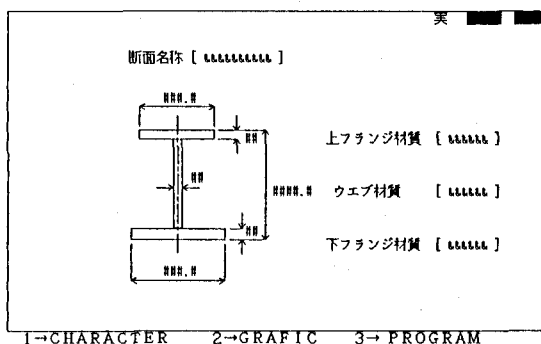
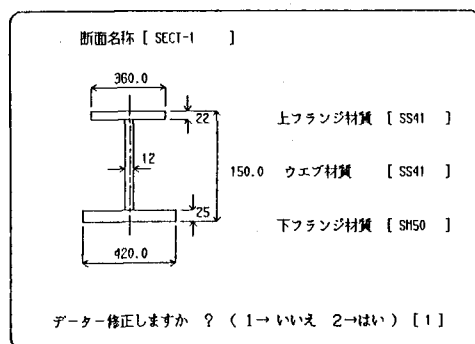
(2) キャラクタモードルーチンでは↑↓←→キー等を用いてカーソルを画面内で移動させ、文字，入出力変数名と、その表示エリアを定義する。また消去，挿入，行換え等の編集機能も備わっている。（４－１参照）

(3) グラッフィックモードルーチンでは、図形の定義を行なう。図形作成コマンドの選択を行なうと画面左上スミに小さな円のマーカが現われ、テンキーを用いてこのマーカを８方向に自由に移動させることができ、その移動速度（ピッチ）も速い，遅いの選択が随時できる。マーカをコマンドに応じて画面上にセットすることにより、所定の図形を定義する。また一度定義した図形の消去、画面拡大状態での図形定義等の機能も有している。（４－２参照）

(4) ＢＡＳＩＣプログラム作成ルーチンでは、画面上の文字，入出力変数とその位置，図形状態等の解析を行ない、プログラムを生成し、アスキーコードでそのプログラムをファイルに収容する。

(5) 修正プログラムのロードルーチンでは(4)で作成したプログラムを変更，修正するために、そのプログラムを読込み、本ツール下の文字、図形状態に再現・表示する。表示された画面を今まで対話処理で作ってきたものとみなして、(2)，(3)により変更，修正ができる。ただし、人手の入ったプログラムについては適用できない。

(6) 入力順変更ルーチンでは(4)で作ったプログラムの入力順番を変更した新たなプログラムの作成を行なう。(4)で作ったプログラムの入力命令に関する順序は画面位置において左から右へ、上から下へ定義された入力変数の順序に従う。図形表示をしながら入力するようなプログラムの場合、上述した入力順では不都合な場合が多いため、本ルーチンで変更する。本ルーチンを通すと、入力変数の一欄表が画面表示されるので、その変数に入力順番号を指定してやれば良い。

(7) ユーザプログラムの画面挿入ルーチンでは、すでに作られているプログラムの画面を本ツール下

図－１　プログラム構成

図－２　円弧の作図

図－３　画面編集の例

図－４　実　行　例

で画面表示することができる。この画面に重ねて、新たなプログラムを作成でき、例えば画面上の一部図形を取換えながら画面表示したいようなプログラムを作成する場合に便利である。

4．コマンドの種類

キャラクタ，グラフィックモードにおけるコマンドの種類は各々，9，22個ある。RETURNキーを押続けると、25行目にコマンドの種類とその選択番号がエンドレスに表示されるので、実行したいコマンド番号を選択し、ワンタッチ（たとえば12なら1，2キーを連続して押すだけで、RETURNキーを最後に押す必要はない）で入力すると、そのコマンドに対するガイダンス情報が25行目に表示される。このガイダンスに従い操作していくと、そのコマンドの実行ができることになる。

4－1．キャラクターモードにおけるコマンド

a. 文字プリント………画面上の任意位置に文字（英数字，カナ文字，英記号，漢字）を定義する。漢字を定義する場合には、カナ漢字変換機能を用いる。ただし、#，&，．，”の4つの英記号についてはシステム上の都合から画面上に定義することは許されない。

b. 変数インプット……入力変数名の定義とその表示エリアの定義を行なう。文字型変数の場合にはBASIC文法にもとずいて変数名の末尾に$記号をつける。数値型変数の場合には、すべて単精度実数とする（変数の末尾に型を示す%，#等をつけてはならない）。文字型変数の場合、&&&&，数値型変数の場合，###．##のように、その入力データの画面表示位置に&，#記号を定義する。（図－3参照）

c. 変数プリント………出力変数名とその表示エリアを定義する。定義のしかたはbと同じである。

d. エディット…………aで設定した文字，b，cで設定した表示エリアに関する消去，挿入，行換え，文字定義が行なえる。b，cの桁数，小数点位置，文字数の変更も&，#，．を画面上で再定義することにより可能である。

e. 変数名変更…………カーソルを#，&の位置に合せることによりガイダンスエリアにその変数名を表示する。表示後、変数名の変更，誤操作による入出力変数の入換えが行なえる。

f. 同一文字……………画面上に同じ文字列をいくつか定義したい場合、その文字列をカーソルで指定すると、画面任意位置に同一文字列をつくることができる。

g. カラム表示…………24行×80カラムの枠表示、消去をコマンド選択ごとに繰返す。

h. 作　表………………セミグラフィックコードを用いた縦・横線から成る表枠をカーソル移動定義により効率よく作成する。

i. シフト………………25行目に文字等を定義したい場合に用いる。このコマンドが選択されると1行分のスクロールが行なわれる（25行目をガイダンスエリアとして用いているため）。

4－2．グラフィックモードにおけるコマンド

a. 直　線………………3．(3)で説明したマーカを2点、画面上に定義することにより、そのマーカを結ぶ直線が定義される。

b. 連続直線……………マーカの連続セットにより、折線を定義する。

c. 四角形………………対角の2点セットによる。

d. 円……………………円中心点と円周上の1点のセットによる。

e. グラフィック文字…文字描画点、描画方向をマーカセットにより定義し、文字の大きさ（縦横倍率）、文字列を入力する。

f. 作　表………………カーソル移動により表の左上スミ、右下スミを定義し、その外側枠の中にやはりカーソル移動により任意長さの横線、縦線を定義できる。このコマンドで作られた枠線はプログラムになった場合にはLINE命令となる。

g.　寸法線………………水平，鉛直寸法線のみ可能である。寸法引出線の足の部分２点と寸法線位置を定める１点の合計３点のマーカセットによる。引出線も同時に定義され、他の寸法線の引出線と重なった場合には、その引出線は定義されない。

h.　円　弧………………２直線にはさまれる円弧，２直線に接する円弧，最初に円を定義しておいてその円周上の２点を始終点とする円弧の３種類。円弧マーカのセット要領を図－２に示す。

i.　消　去………………ａ～ｈ（ｆは除く）で定義した図形の消去。消去したい図形上にマーカセットを行なう（グラッフィック文字の場合は文字大略原点位置）。マーカ円内に図形が入らないと、その図形は検索されない。またマーカ位置に２つの図形が重なった場合、後で定義した図形の方が消去される。

j.　拡　大………………対角の２点のマーカセットによってできる四角形内に入っている図形を画面一杯に表示する。この状態でａ～ｉ（ｆは除く）のコマンドを選択できる。ただし、キャラクタモードで定義した文字等は拡大されず、画面表示からもはずされる。

k.　原図復帰……………拡大モードの消去を行なう。

l.　再表示………………マーカ走行によってつぶされた図形をきれいに再表示する（マーカのオン，オフによってマーカにかかった図形の一部が消去されるため）。

m.　線種設定……………実線，点線，１点鎖線，２点鎖線の４種類。現在設定されている線種は画面上右スミに常に表示されている（図－３参照）。たとえば、１点鎖線の状態で円を定義すると、１点鎖線の円が定義される。

n.　カラー設定…………図形の設定色，ペイントをする際の領域色の２色の設定。設定されている２色は画面上スミに矩形表示されている（カラーの場合）。

o.　ペイント……………領域内の１点のマーカセットによりｎで設定した領域色で、図形設定色の境界内をペイントする。

p.　図形移動……………対角の２点のマーカセットによりできる四角形内に入っている図形を、画面任意位置に平行移動する。

q.　図形コピー…………対角の２点のマーカセットによりできる四角形内に入っている図形を、画面任意位置にコピーする（同じ図形を同一画面内に複数定義したい場合に用いる）。

r.　文字マーカ…………キャラクタモードにおける１文字分の文字の大きさを示す矩形マークをカーソル移動により設定できる。拡大状態でこの矩形マークも拡大表示されるので、文字位置との兼合をみながら微細な図形の定義が可能である。ただし、矩形マークは作られるプログラムとは無関係である。

s.　カラム表示…………２４行×８０カラムの枠の表示，消去。拡大時には、枠も拡大される。

t.　点表示………………定義された直線の両端，円の中心，円弧の中心と始終点には必ず点が定義されるようになっている。このコマンドを選択すると、このような点の表示，消去が交互に繰返される。点はマーカと同じ大きさの円で表示される。

u.　厳密直線……………点表示状態にして表示点にマーカをセットすると、その２点を結ぶ厳密な直線を定義できる。これによりペイントの色モレ等を防止できる。

v.　フィレット…………定義直線の一部を除去する。円弧定義（図－２参照）等により不要となった直線の一部を除去する。

なお、画面上に定義できる図形の数に制限がある。画面を縦方向に１０等分割し、その各々のゾーンに対して定義できる図形数は５００までである。

５．作成例と使用法

図－３にプログラム作成直前の画面ハードコピー，図－５に生成されたプログラム，図－４にこのプログラムを実行した際のハードコピーを簡単な作成例として示す。

プログラムは分番号１００から始まって、１０ステップごとに作られているが、長くなるので途中、省略してある。図－３の画面に入力変数，点線・１点鎖線・２点鎖線の図形，寸法線が定義されている場合、図－５のプログラムを実行する際、専用のサブルーチンと一緒に実行する必要がある。

サブルーチン６５２００（ステートメント１５０）は線種，カラー番号等を指定して破線の直線，矩形を描画するルーチンである。同様に円，円弧のためのサブルーチン６５３００が用意されている。６５４００（１９０－２１０）は縦横の区分，矢印の有無等を指定して寸法線，寸法引出線を描画するルーチンである。６５０００（４７０以下）は文字型・数値型の区分，文字数・桁数・小数点位置，カーソル位置等を指定して入力データの汎用的処理を行なうルーチンである。６５０００～６５４００はメモリ上に常駐させて使用する。

文番号３９０は本ツールの標準形として作られ、この命令を変更したり、作成プログラムの外側に出して使用することによって、入力データに関する処理手順を自由に変えることができる。ＮＬＯＯＰは処理の数，文番号４１０のＭＯＤＥは１→キーイン入力，２→入力値の画面表示，３→入力データ修正，４→ファイル書き込み，５→ファイル読込みである。標準形の場合、キーイン入力（キーイン入力した時は入力値の画面表示機能がある），修正（文番号５９０ではいと答えた時のみ修正ルーチンに入る）の形となる。

ＮＬＯＯＰ以下をプログラムの外に出し、ＮＬＯＯＰ＝３：ＭＯＤＥＴＰ（１～３）＝１，３，４：ＧＯＳＵＢ１００とするとキー入力，修正，ファイル保存の手順で処理され、ＮＬＯＯＰ＝４：ＭＯＤＥＴＰ（１～４）＝５，２，３，４：ＧＯＳＵＢ１００とするとファイルからの読出し，表示，修正，ファイル保存の手順で処理される。作られたＳＵＢ１００は処理手順によらず共通に使用できる。

文番号３９０のＮＤＡＴＷ，ＰＡＧ＄はインデックスシークエンシャルファイルのキー設定に用いる。ＰＡＧ＄はアプリケーションに現われる画面名称でデータケースを示す名称ＭＥＩＳＹＯ＄との合成によりキーの頭部名を決定する。ＮＤＡＴＷはその画面に現われる入力変数のカウント値でＳＵＢ６５０００内でそのカウントを行ないユニークなキーの尾部を決定する。インデックスシークエンシャルファイルはデータ数がが多くなるとキー検索に時間がかかるので、アプリケーション作成にあたっては十分に検討する必要がある。

本ツールの性格上、画面単位のデータ管理が行なえるインデックスシークエンシャルファイルを用いざるをえなかったが、ファイルとしてはシークエンシャル，ダイレクトファイルを用いた方が便利な場合が多い特にランダムファイルはレコードの管理を行なわなければならないので、そのプログラムは面倒なものとなる。そのため、ダイレクトファイルへのＩ／Ｏを総括的に行なうことのできるサブルーチンを開発し、それを利用している。

作成されたプログラムは上述したように３９０の命令を使用目的に合せて調整するだけで、すべての機能を発揮することができる。

６．実用上の効果

本ツールを用いることで入出力部分が定型的なものであれば、その開発の手間は１／３～１／５となった。これは、グラフィック命令を用いる場合、めんどな座標値のひろい出し、座標計算といったものが不必要で、自動生成されるプログラムには、まったく誤りがないからである。また作られるプログラムは非常に定型的なものになっており、プログラムの標準化という意味でメンテナンス等に際して効果を発揮している。今後繰返し、分岐等の機能を追加できないか、検討していくことにしている。

なお、本ツールの使用言語はすべてＢＡＳＩＣで、使用機種はユニバックのＵＰ１０Ｅである。

```
100 '--------------------------------------------------
110 'SAMPLE PROGRAM
120 '--------------------------------------------------
130 VIEW (0.0)-(639.399):WINDOW (0.0)-(639.459)
140 REM : GRAFIC START
150 SENSYU-3:SIRO-1:SBOX-0:SXS- 202:SYS- 144:SXE- 202:SYE- 334:GOSUB 65200
160 LINE ( 156.1. 164)-( 244. 176). 1.B
190 SYOTA-1:SYATP-1:SIRO- 1:SXS- 156.1:SYS- 132:SXE- 244:SYE- 132:GOSUB 65400
200 SYOTA-2:SYATP-0:SIRO- 1:SXS- 156.1:SYS- 158:SXE- 156.1:SYE- 132:GOSUB 65400
210 SYOTA-2:SYATP-0:SIRO- 1:SXS- 244:SYS- 158:SXE- 244:SYE- 132:GOSUB 65400
370 SYOTA-1:SYATP-2:SIRO- 1:SXS- 206.2:SYS- 244:SXE- 226.2:SYE- 244:GOSUB 65400
380 REM : GRAFIC END
390 NDATW-0:PAG$-"1":NLOOP-2:MODETP(1)-1:MODETP(2)-3
400 FOR SINDEX-1 TO NLOOP
410 MODE-MODETP(SINDEX)
420 ON MODE GOTO 430 ,430 ,470 ,470 ,470
430 LOCATE 19.4:PRINT "断面名称 [             ] ":
440 LOCATE 48.10:PRINT "上フランジ材質  [        ] ":
450 LOCATE 46.14:PRINT "   ウェブ材質    [        ] ":
460 LOCATE 48.18:PRINT "下フランジ材質  [        ] ":
470 MTYP-2:MOJISU-10:CX-30:CY-4:IF MODE-1 THEN GOSUB 65000:NAM$-MOJI$ ELSE MOJI$
-NAM$:GOSUB 65000:NAM$-MOJI$
480 MTYP-1:MOJISU-5:KETASU-1:CX-24:CY-7:IF MODE-1 THEN GOSUB 65000:WUF-SUJI ELSE
 SUJI-WUF:GOSUB 65000:WUF-SUJI
490 MTYP-1:MOJISU-2:KETASU-0:CX-36:CY-10:IF MODE-1 THEN GOSUB 65000:TUF-SUJI ELS
E SUJI-TUF:GOSUB 65000:TUF-SUJI
500 MTYP-2:MOJISU-6:CX-66:CY-10:IF MODE-1 THEN GOSUB 65000:ZAIU$-MOJI$ ELSE MOJI
$-ZAIU$:GOSUB 65000:ZAIU$-MOJI$
510 MTYP-1:MOJISU-2:KETASU-0:CX-28:CY-13:IF MODE-1 THEN GOSUB 65000:TWEB-SUJI EL
SE SUJI-TWEB:GOSUB 65000:TWEB-SUJI
520 MTYP-1:MOJISU-6:KETASU-1:CX-40:CY-14:IF MODE-1 THEN GOSUB 65000:HWEB-SUJI EL
SE SUJI-HWEB:GOSUB 65000:HWEB-SUJI
530 MTYP-2:MOJISU-6:CX-66:CY-14:IF MODE-1 THEN GOSUB 65000:ZAIW$-MOJI$ ELSE MOJI
$-ZAIW$:GOSUB 65000:ZAIW$-MOJI$
540 MTYP-1:MOJISU-5:KETASU-1:CX-24:CY-20:IF MODE-1 THEN GOSUB 65000:WLF-SUJI ELS
E SUJI-WLF:GOSUB 65000:WLF-SUJI
550 MTYP-1:MOJISU-2:KETASU-0:CX-36:CY-17:IF MODE-1 THEN GOSUB 65000:TLF-SUJI ELS
E SUJI-TLF:GOSUB 65000:TLF-SUJI
560 MTYP-2:MOJISU-6:CX-66:CY-18:IF MODE-1 THEN GOSUB 65000:ZAIL$-MOJI$ ELSE MOJI
$-ZAIL$:GOSUB 65000:ZAIL$-MOJI$
570 IF MODE-1 OR MODE-3 THEN 590
580 IF MODE-2 AND SINDEX<NLOOP AND MODETP(SINDEX+1)-3 THEN 590  ELSE 640
590 LOCATE 15.25:PRINT "データー修正しますか  ? ( 1→ いいえ  2→はい ) [
 ]  "::LOCATE 70.25:INPUT:"".OKSW
600 IF OKSW<1 OR OKSW>2 THEN 590
610 IF OKSW-1 THEN 620  ELSE 630
620 IF MODE-1 OR MODE-2 THEN SINDEX-SINDEX+1:GOTO 640  ELSE 640
630 IF MODE-1 OR MODE-2 THEN 640  ELSE 420
640 NEXT SINDEX
650 RETURN
```

図－5　作成されたルーチンの例